Re 09/241,823 -- EAST 1.2

Type	L #	Hits	Search Text	DBs	Time Stamp
BRS	L1	29	broadcast$3 same (updat$3 or upgrad$3)same push$3	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 09:56
BRS	L2	12190	auto or automatic$4)near5(restor$3 or recover$3 or backout or back adj1 out	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 10:03
BRS	L3	1	l1 and l2	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 10:03
BRS	L4	2322	install$5 same(updat$3 or upgrad$3)same(software$1 or program$4)	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 10:07
BRS	L5	1217	(distribut$3 or broadcast$3) and l4	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 10:08
BRS	L6	33	l2 and l5	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 11:42
IS&R	L7	1	("717/11").CCLS.	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 11:40
IS&R	L8	502	("717/168-178").CCLS.	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 11:40
IS&R	L9	3209	("709/203,217-222").CCLS.	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 11:41
BRS	L10	48	l2 and l4	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 11:42
BRS	L12	46	l2 and l9	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 11:42
BRS	L11	10	l2 and l8	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/01/18 11:46

DOCUMENT-IDENTIFIER: US 6338149 B1
TITLE: Change monitoring system for a computer system

BSPR:
The process of upgrading computers to a new level of operating system can take days, and typically requires some level of verification and validation of the installed software products. This process can prove quite costly in terms of manpower and system availability.

DEPR:
The various template distribution functions are supported by dialogs that: (1) distribute a template to a node and, thus, create duplicate platforms; (2) construct a template from a set of available products; and (3) change owner, group, and permissions for multiple products (e.g., which is of import for the initial distribution of the products). The second construct template function may: (a) permit each monitored node to have a unique set of products; (b) change the products included in the template for a node (e.g., add products, remove products); and (c) change schedules and e-mail notification.

CLPR:
46. The change monitoring and response system of claim 44 wherein said means for restoring includes means for automatically restoring said at least one of said products.

DOCUMENT-IDENTIFIER: US 6304882 B1
TITLE: Data replication system and method

BSPR:
The present invention relates to design of distributed data systems and methods, and more particularly, to design of data replication systems and methods.

BSPR:
Distributed data systems (and methods) use a central database with copies of that database distributed to client computers within the system. For example, in a conventional distributed data system having a central server computer and one or more client computers, each client computer uses a copy of the central database or data repository that is located on the server computer. Each client computer performs computer application functions and operations using the copy of the database. To keep each copy of the database at the client computer matching with the central database located at the server computer, conventional distributed data systems use conventional data replication systems.

BSPR:
Another problem with using locks to serialize transactions against different copies of a database is that if a lock is visible over an unreliable network, very difficult failure situations arise, such as network partitions. Moreover, if the server database is no longer in agreement with the copies of the database at the client databases, there is an increased probability that the data in the distributed data system may become compromised. Once the data is compromised, the system fails. Thus, conventional data replication systems allowing both "read" and "write" transactions are not suitable for mission critical distributed data systems where maintaining data integrity is essential.

BSPR:
A sixth problem with conventional data replication systems is that they do not typically automate the distribution aspects of a computer software upgrade such that the client computer remains operable and the database useable. Many existing data replication systems require new software or upgrade utility installations on every client computer. During the installation process, the client computer must remain unusable so that the database is not corrupted. To be successful, the installation must be well planned, including recovery plans in the event that the upgrade fails. The installation must also be well tested

and run at times when the client computers are least used so that the process is least disruptive. Thus, existing data replication systems are unwieldy, especially in large installations, for example having 10,000 client computers and are not well suited in increments requiring a high degree of client computer availability.

DEPR:
A preferred embodiment of the present invention will be described with reference to the Figures, where like reference numbers typically indicate identical or functionally similar elements. The present invention includes a system and a method for data replication in a distributed environment.

DEPR:
The replication protocol of the present invention also allows for fast, efficient disaster recovery, for example, when the source database recovers from a backup and loses some recent transactions. The replication protocol automatically restores agreement for the target databases of each target database system 150 without requiring separate database administration.

DOCUMENT-IDENTIFIER: US 6295575 B1
TITLE: Configuring vectors of logical storage units for data storage partitioning and sharing

DEPR:
With reference to FIG. 1 of the drawings, there is shown a cached storage subsystem 20 connected via a data network 21 to a plurality of hosts 22, 23, 24, 25. The cached storage subsystem 20 includes storage volumes 26 and a storage controller 27 for controlling access of the hosts to the storage volumes. The storage volumes are logical units of storage distributed over one more storage devices 28, 29, 30, and 31. The storage devices are magnetic disk drives, optical disk drives, tape drives, solid-state memory devices, or other storage devices capable of providing nonvolatile data storage. Presently the preferred storage devices are magnetic disk drives each having a storage capacity of at least 46 gigabytes.

DEPR:
The data network 21 could also include one or more additional cached storage subsystems, preferably at different geographical locations, to provide redundancy and protection from a catastrophic failure of the cached storage subsystem 20, as will be further described below with reference to FIGS. 39 and 40. The cached storage subsystems could be linked for automatic remote mirroring of data to provide recovery from a disaster, as described in Yanai et al., U.S. Pat. No. 5,544,347 issued Aug. 6, 1996, incorporated herein by reference, and in Yanai et al., U.S. Pat. No. 5,742,792 issued Apr. 21, 1998 (Ser. No. 08/654,511 filed May 28, 1996), incorporated herein by reference.

DEPR:
Referring to FIG. 16, there is shown a flowchart 160 of a microcode routine executed by the microprocessor (76 in FIG. 4) of the port adapter (35 in FIG. 4) for volume configuration when installing the storage subsystem into the data processing system of FIG. 1. In a first step 160, the system administrator enters a group name for a host to be permitted access through one or more port adapters. For example, the system administrator manually operates the keyboard 92 in FIG. 4 to enter the host name, host controller number(s), and port adapter number(s). The service processor 93 forwards this information to the port adapter(s) and the microprocessor(s) in the port adapter(s) allocate(s) a table entry for each host controller number and places the respective group name into each table entry. Then in step 162, the port adapter(s) automatically search the network for the S_ID and WWN of the host controller ports corresponding to the group names. For example, the port adapters poll the ports of the hosts, and the host controllers are programmed to respond to port adapters by returning their host names and controller numbers. If the polling is successful in returning the S_ID and WWN associated with the group name, as tested in step 163, then execution continues to step 164. In step 164 the system administrator enters a volume list for the group name, for example by entering logical volume numbers, ranges of logical volume numbers, or vector specifications (beginning number, ending number, stride) at the keyboard 92 in FIG. 4. In step 165, the microprocessor creates a table entry for the group name and its associated S_ID,

WWN, and a pointer to the volume list. In step 166, execution continues to step 167 if the system administrator has no more group names to enter; otherwise, execution loop back to step 161. In step 167, the volume access table and volume lists are copied to a storage volume as back-up for port adapter error recovery or diagnostic purposes, and the routine if finished. The back-up copy should be updated if additional group names are added to the volume access table or if the volume lists are changed

DOCUMENT-IDENTIFIER: US 6259442 B1
TITLE: Downloading software from a server to a client

BSPR:
A problem encountered by computer users in general is that software applications tend to become outdated quickly. Accordingly, software suppliers periodically produce upgrades, which are often distributed in the same way that the original software was distributed, such as on magnetic or optical disks or other similar storage devices. However, the distribution of software upgrades on storage media such as these has disadvantages. For example, it is inconvenient and sometimes annoying for the user to have to repeatedly install software upgrades, which can be a time-consuming process. Further, a user may not be aware that an upgrade is available or necessary, or the user may forget to obtain or install the upgrade . The failure or delay in installing an upgrade can be detrimental since the upgrade may add valuable new features to the software or remedy a "bug" (i.e., error) in the software . Therefore, what is desired is a technique for allowing a software upgrade to be automatically provided over a network in a manner which requires little or no effort on the part of the user.

BSPR:
The present invention relates to upgrading or restoring software stored on a client, such as a computer or set-top box. In one implementation of the invention, inoperable or corrupted software at the client is automatically restored. During power up or at any other predetermined time, the client automatically checks whether the software or data is corrupt or valid (i.e., not corrupt) using, for example, a conventional checksum technique. If the client determines that the software or data is corrupt, the client automatically connects to a server that contains a replacement for the corrupt software or data. Then, the replacement is downloaded over the network infrastructure to the client so that the corrupt software or data can be replaced by the valid software or data, respectively. Thus, corrupt software or data can be automatically fixed with little, if any, effort required of the user.

DEPR:
Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including set-top boxes, personal computers, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by local and remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

DEPR:
Thus, various methods according to the invention for upgrading or restoring software stored at the client have been disclosed. Corrupted or inoperable software can be restored automatically when a client is powered on by receiving replacement software from the server. Outdated software can be replaced at any time by receiving at the client a download of a software upgrade via a satellite link. The methods of restoring corrupted software and upgrading outdated software can be practiced in combination or separately as desired.

CLPR:
1. In a computer system that includes a plurality of client systems and a plurality of server systems, all interconnected by a network infrastructure, wherein the plurality of server systems provide access to sites storing Web pages or other data, and wherein one or more of the client systems comprises a

conventional television monitor and set-top box having a processing system that includes program instructions used in the operation of the one or more client systems, such as in retrieving data from said sites, a method of restoring a corrupted portion of the program instructions at the client system, wherein the restoring occurs automatically and without user intervention, the method comprising the client system performing the following steps:

CLPR:
11. A computer program product for implementing, in a computer system that includes a plurality of client systems and a plurality of server systems, all interconnected by a network infrastructure, wherein the plurality of server systems provide access to sites storing Web pages or other data, and wherein one or more of the client systems comprises a conventional television monitor and set-top box having processing system that includes program instructions used in the operation of the one or more client systems, such as in retrieving data from said sites, a method of restoring a corrupted portion of the program instructions at the client system, wherein the restoring occurs automatically and without user intervention, the computer program product comprising:

DOCUMENT-IDENTIFIER: US 6163859 A
TITLE: Software vault

BSPR:
Inexorable advances in electronics have led to widespread deployment of inexpensive, yet powerful computers that are networked together. Over time, programs installed on these computers are updated and these updates need to be maintained. Information system departments and their users face the thorny task of maintaining numerous instances of software across their complex distributed environments. The maintenance process covers a number of tasks, including software installation, synchronization, backup, recovery, analysis and repair.

BSPR:
Historically, relationships between software components have been manually detected and component states have been recorded in a log. This state information is external of the components themselves and must be updated whenever the components change. As state information is recorded only at the time of installation, changes made subsequent to installation may be lost. As the rate of change increases and complexity of the software configuration grows, the external state representation becomes difficult to maintain and prone to error. Moreover, during normal operation, users may make changes to the software through individual personalization and through the installation of additional software, thus changing the state information. The difference in state information between software installation and software operation can lead to unpredictable behavior and may require support from information system personnel.

BSPR:
Moreover, the vault can be used to diagnose problems by comparing an existing state on a client computer to both a previously working state and a reference state stored in the vault. Further, the vault can be used to allow applications which have been damaged to self-heal applications by automatically restoring previously working states or reinstalling components from reference states.

BSPR:
Further, the vault decreases network overhead and increases scalability of electronic software distribution by eliminating delivery of duplicate files that make up software packages. The flexible architecture of the invention protects user investment in existing solutions for enterprise-wide systems management, network management, and application management.

DEPR:
Turning now to FIG. 12, the installation step 470 (FIG. 8B) is shown in more detail in FIG. 12. First, the vault catalog is loaded (step 472). Next, the highest version of the software stored in the vault is determined (step 474). The metadata associated with the highest version of the software is copied to

the target machine (step 476). Further, data is remapped (step 478). The process of FIG. 12 then applies a preprocessing operation to the remapped data (step 480) to convert data into the proper format and set up variables appropriately, among others. Further, items associated with the software are installed (step 482). A post-processing process is applied (step 484). This step is similar to step 480 in that variables are checked and data is formatted. Finally, an inventory of the software being installed is updated (step 486) before the process exits (step 488).

DOCUMENT-IDENTIFIER: US 5978594 A
TITLE: System for managing computer resources across a distributed computing environment by first reading discovery information about how to determine system resources presence

ABPL:
A method and apparatus are disclosed for managing a computer network. A manager software system is installed on a network management computer system within the network, and one agent software system is installed on each of the server computer systems in the network. A knowledge module in the form of a text fie is stored on the network manager computer system so that the manager software system can transmit knowledge to the various agent software systems throughout the network, for use by the agents in monitoring and managing the server on which they are installed. Interpretable script language programs are present on all computers in the network, expanding and customizing the functionality of the agent software systems. A method is disclosed for using the high level interpretable script language programs in connection with the agent software systems for discovering resources on the network, monitoring aspects of resources, and taking recovery actions automatically in the event of an alarm condition.

BSPR:
This invention relates generally to computer networks. More specifically, the invention relates to a method and apparatus for centrally monitoring and managing the computers, applications and other resources present in a distributed computing environment.

BSPR:
The data processing resources of business organizations are increasingly taking the form of a distributed computing environment in which data and processing are dispersed over a network comprising many interconnected, heterogeneous and geographically remote computers. Among the reasons for this approach are: to offload non-mission-critical processing from the mainframe; to provide a pragmatic alternative to centralized corporate databases; to establish a single computing environment; to move control into the operating divisions of the company; and to avoid having a single point of failure. For example, many business entities have one client/server network installed in each regional office, in which a high-capacity computer system operates as the server supporting many lower-capacity desktop computers. The servers in such a business entity are also commonly connected to one another by a higher-level network known as a wide area network. In this manner, users at any location within the business entity can theoretically access resources present in the company's network regardless of where the resource is located.

BSPR:
The invention is a novel method and apparatus for managing a computer network. The method is intended to be used in any distributed computing environment in which two or more computer systems are connected by a network, including environments in which the networked computers are of different types. A manager software system is installed on and runs on one of the networked computer systems designated as the network management computer system. The network management computer system acts as a central console for managing the entire network. An agent software system is installed on and runs on each of the other computer systems in the network. Each respective agent software system carries out tasks on the computer system in which it is installed, such as discovering which resources and applications are present on that computer system, monitoring particular aspects of the resources and applications present on that computer system, and executing recovery actions automatically when such actions are warranted. Each agent also carries on a dialog of communication with the manager software

system via the network, so that the central console on the network management computer system can provide a continuously updated display representing all resources and applications present throughout the network, as well as the state of each such resource or application. A knowledge module is installed on the central network management computer system, and then that computer system distributes knowledge throughout the network to the agent software systems. Such knowledge allows the agent software systems to operate in a semi-autonomous and automatic manner, relieving burdens from the central console operator. Also, the agent software systems make use of script programs written in a high-level interpretable language in order to execute certain procedures. Thus, only one agent software system need be installed on each server. As needs for agent software functionality changes or increases over time with changes in the network, new script programs may be developed and distributed throughout the network to customize the agent software systems instead of replacing, modifying or duplicating the agent software systems themselves.

DOCUMENT-IDENTIFIER: US 5499357 A
TITLE: Process for configuration management

BSPR:
The introduction of networks and decentralized processing environments adds a significant level of complexity to managing system configurations across distributed systems. In distributed architectures, system components are not confined to a single, isolated processing environment. Compatibility of system components and integrity of the environment require verifying compatibility and ensuring integrity across multiple environments running on various processing platforms.

BSPR:
Work station technology, networks and client-server architectures introduce additional complexities in configuration management. The cooperative nature of these distributed system architectures, having multiple remote locations with each location operating independently of another location and having separate system configurations, presented skilled artisans with complex, unresolved configuration management issues.

BSPR:
In an attempt to address these new issues in configuration management, presented by the more contemporary distributed computer processing systems, Dean et al. developed a procedure for classifying areas subject to and amenable to automated computer support. In their distributed system, configuration and structure of the distributed system modeled is described by defining structures, relations and components of a distributed system.

DEPR:
Ensuring integrity of the defined system configuration may occur at various, pre-selected times. For example, verifying and maintaining defined configuration relationships and inter-relationships may occur when a new system component is installed . Adding a new fixed disk drive requiring enhanced-function operating software exposes a defined system configuration to a potential operating error if the relationship of the components are not verified. Thus, verifying and maintaining the integrity of the configuration is necessary after upgrading the disk drive. Subsequent verification and maintenance checkpoints may be required depending on the significance of the relationship to be validated and impact to the system if a relationship becomes corrupted between checkpoints. Other exemplary opportunities for verification of configuration integrity include 1) after an initial installation of a complete system, 2) installation of one or more components in a system, 3) a partial or complete downgrade of one or more system components (e.g., such as a hardware device or software application) to a previous version of a system component, or 4) a partial or complete upgrade of one or more system components to a later component version.

DEPR:

Finally, incompatibilities identified in the integrity validation are eliminated. Although alternatives exist for identifying incompatibilities, a preferred method reads migrational software and resident software identification data, verifies compatibility between resident software and migrational software by comparing current configuration data against the dedicated control field, and upon verification, updates current configuration data for the migrational software, now "new" resident software, in the computer system. Preferable computer operations are not limited to migrational software installation, but may include a software upgrade or downgrade.

DEPR:
The installed document production facility becomes the current configuration, having established hardware components, a particular software version and release for installed applications and designated user languages, and is designated as such, 104. Upon obtaining an initially configured system, system PROMS are updated and retain current configuration information for subsequent use in validating integrity of the system configuration, 106.

DEPR:
Upon analysis of the control data retrieved from application files or read from system data stores, the CM application instructs InstallFile to load the software files from tape I to internal fixed storage (disk) on the CPU A. Simultaneously, InstaliFile updates permanent system files with current configuration control data. A full software upgrade is completed and the integrity of the system is ensured. Upon successful completion of the full software upgrade, the system software label is updated (via SWLabels) and status information, reporting on the integrity of the upgraded current configuration, is provided to users through the system application Dialog (FIG. 4).

CLPR:
9. The process according to claim 8, further comprising the step of performing a computer operation that is at least one of an installation of migrational software or an upgrade or downgrade to resident software .

CLPR:
11. The process according to claim 8, wherein the step of automatically taking corrective action includes restoring the resident software on the automated computer system to a pre-operation configuration in response to an identified incompatibility.

DOCUMENT-IDENTIFIER: US 6256773 B1
TITLE: System, method and article of manufacture for configuration management in a development architecture framework

DEPU:
The backup product should have fundamental management capabilities. Automatic restore, unattended operation and command line processing of the product should be available. Basic tape functions such as cataloging, internal labeling, initialization, certification, scratch protection and write protection are musts.

DEPU:
Pre-defined installation and de-installation scripts. Ability to perform complete back-out of all related segments quickly and automatically, without impacting other, successfully installed updates.

DOCUMENT-IDENTIFIER: US 6189147 B1
TITLE: Apparatus and method for an installation recovery system

BSPR:
Heretofore, installation managers have lacked the ability to automatically recover an install process in the event of a crash without wasting time re-installing files that are already installed on the user system.

DOCUMENT-IDENTIFIER: US 5715462 A
TITLE: Updating and restoration method of system file

BSPR:
Next, the file replacing function of the activated OS of the second memory area is called so as to replace the plurality of system files in the first memory area with the substitute files prepared in advance. That is, the file contents are updated. Further, the backup files of the replaced system files are prepared. Thereafter, it is determined whether or not the file replacement is finished normally, if not finished normally, the foregoing backup files are restored in the first memory area. That is, the state prior to the replacement (prior to updating) is returned. If the file replacement is finished normally or if the restoration of the backup files is finished, the OS in operation of the second memory area is terminated and the OS of the first memory area is activated. By this, the automatic updating process and the automatic restoring process at the time of update failure, of the system files which are access-locked are performed.

DOCUMENT-IDENTIFIER: US 5713024 A
TITLE: Cold boot data backup system

ABPL:
The cold boot data backup apparatus maintains an index of all data file activity on a computer system and stores copies of data files in a manner to enable a user to recreate the state of the computer system at any selected point in time. This apparatus automatically formats the computer system memory in response to a failure thereof and automatically restores the operating system, all application programs and every data file that is selected by the user to be monitored and preserved by this apparatus.

BSPR:
This invention relates to a data backup system that automatically produces a temporal record of all data file activity in a computer system and provides apparatus to record all activity for user identified data files, which apparatus automatically configures and restores the computer system memory upon a failure of the memory.

CLPR:
1. A cold boot data backup apparatus for automatically restoring data files stored in a memory of a computer system in response to a failure of said memory, using a backup media that is connected to said computer system which has stored thereon a copy of data files written in said memory, every version of data files changed by said computer system, comprising:

DOCUMENT-IDENTIFIER: US 5623661 A
TITLE: System for and method of providing delta-versioning of the contents of PCTE file objects

ABPL:
A method and system for providing delta-versioning of data stored in an object-based data repository. In the preferred embodiment, delta-versioning is provided for "file" objects in a PCTE implementation in a manner which is relatively versatile for developers of PCTE compliant tools and programs and which is largely transparent to the users of those tools and programs. Further, redundant restoration of delta-versioned data is minimized and automatic removal restored data which is no longer needed is provided.

DEPR:

An additional feature of the present invention is that of shared storage for restored contents files and automatic clearing of file system storage space when the restored contents are no longer needed. When a read operation (invoked by a "contents.sub.-- open" or similar API) is performed on a "file" object whose contents have been delta-versioned, a check is performed to see if any other user already has restored the desired contents. Specifically, a check is performed to see if the restored contents file is already present on host file system 40. If so, the user is provided with a file handle to that restored contents file. If no other user is accessing the desired contents (i.e.--the restored contents do not already exist on the host file system), the contents are restored by the delta-versioning program (as described above) to a file on the host file system and the user is supplied with the file handle of the restored file while the delta control file and associated delta remain unchanged.

DEPR:
The present invention provides for the delta-versioning of data associated with objects, such as the PCTE "file" object, in a manner which is transparent to the user and yet flexible for the implementer of the tool or program. The present invention also provides for the automatic freeing of storage space occupied by restored versions of contents files which are no longer required.

DOCUMENT-IDENTIFIER: US 5581749 A
TITLE: System and method for maintaining codes among distributed databases using a global database

DEPR:
The error management module 524 is invoked by any other module upon the occurrence of an exceptional error such as one caused by a communication malfunction, a database error or a program logic error. The invoking module sends the error management module 524 a code which identifies the error. The error management module then takes a snapshot of the system-specific information that would help isolate the cause of the error, and then logs the error and the snapshot data to the abnormal events audit log. Depending on the nature of the error, the error management module 524 may then attempt to recover from the error without human intervention. Such an attempt, as well as its results, are logged to the abnormal events audit log 525. An administrative user periodically examines the abnormal events audit log 525 and may attempt to recover from any errors not already automatically corrected by the error management module 524. In addition to facilitating the recovery of exceptional errors, the abnormal events audit log 525 indicates any recurring patterns of exceptional errors. Such information can be used to identify bugs in the global code system 410, malfunctions in the transaction processing environment, and ways to improve robustness.

DEPR:
Otherwise, the error management system 524 carries out steps 2518-2522 as follows. In step 2518, the error management system 524 attempts to automatically recover from the error. For example, if the error was an unsuccessful distribution due to a communication line which is down, the attempt may involve sending the distribution package over an alternate communication line. Alternatively, if the error was an unsuccessful distribution due to a distribution target which is currently not available, the attempt may involve invoking the destination server 2318 to repeatedly attempt to send the distribution package at specified time intervals until distribution is successful.

DEPR:
Assuming for illustration that the distribution of step 2616 was not distributed to the U.S. inventory system 112 because of a bad communication line, the error management module 524 is then invoked with the error code generated. The error management module 524 then: (1) captures system specific information that may assist in recovering from the error, (2) logs the error code and the system specific information in the abnormal event log 525, and (3) invokes the distribution module 520 to send the distribution package to the U.S. inventory system 112 via an alternate communication line. Assuming this transmission is successful, the error management module 524 then receives an acknowledgement from the U.S. inventory system 112 and updates the entry in the abnormal event log 525 so that the entry indicates that the error has been automatically recovered from (see step 2618).